CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2007; **19**:583–592
Published online 10 October 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.1120

Cross-organization interoperability experiments of weather and climate models with the Earth System Modeling Framework

Shujia Zhou[1,*,†,‡], V. Balaji[2], Carlos Cruz[1,‡],
Arlindo da Silva[1], Chris Hill[3], Erik Kluzek[4],
Shep Smithline[2], Atanas Trayanov[1] and Weiyu Yang[5]

[1]*NASA Goddard Space Flight Center, Greenbelt, MD 20771, U.S.A.*
[2]*Princeton University, 201 Forrestal Road, Princeton, NJ 08450-6646, U.S.A.*
[3]*Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, U.S.A.*
[4]*National Center for Atmospheric Research, 1850 Table Mesa Drive, Boulder, CO 80305, U.S.A.*
[5]*National Center for Environmental Prediction, 5200 Auth Road, Camp Springs, MD 20746, U.S.A.*

SUMMARY

Typical weather and climate models need a software tool to couple sub-scale model components. The high-performance computing requirements and a variety of model interfaces make the development of such a coupling tool very challenging. In this paper, we describe the approach of the Earth System Modeling Framework, in particular its component and coupling mechanism, and present the results of three cross-organization model interoperability experiments. Copyright © 2006 John Wiley & Sons, Ltd.

Received 3 October 2005; Revised 14 July 2006; Accepted 20 July 2006

KEY WORDS: framework; weather model; climate model; model coupling; component

INTRODUCTION

A physical Earth system model typically consists of several model components, which are coupled through exchanging data. For example, the well-publicized El Niño–Southern Oscillation (ENSO) phenomenon is the outcome of atmosphere–ocean interaction, and is usually modeled by coupling atmosphere and ocean components. Most models solve partial differential equations on a large gridpoint set for a long time period and consequently require significant amounts of high-performance

*Correspondence to: Shujia Zhou, NASA Goddard Space Flight Center, Greenbelt, MD 20771, U.S.A.
†E-mail: szhou@pop900.gsfc.nasa.gov
‡Employee of Northrop Grumman Corporation.

computing resources. Since Earth system model components, such as atmospheric general circulation models, also contain many physical processes (radiation, cloud formation, precipitation, etc.), considerable time and manpower are needed to develop a production-quality model. As more physical processes are added to the models, the models are becoming increasingly complex and difficult to modify and maintain. It is even more challenging to compare models from different organizations and to construct new models from model components developed in different organizations. A software framework to facilitate model development and coupling would be a great benefit to the weather and climate communities.

Although there were a number of scientific software frameworks, such as POOMA, Overture, Cactus, the Common Component Architecture (CCA), and PAWS [1–3], the weather and climate community concluded that those frameworks are not adequate for its applications. Therefore, NASA's Earth Science Technology Office/Computational Technologies Project has funded the development of the Earth System Modeling Framework (ESMF). The ESMF project enables close collaboration from major U.S. Earth system modeling organizations: the NSF National Center for Atmospheric Research (NCAR), the NASA Goddard Space Flight Center Global Modeling and Assimilation Office (GMAO), the Massachusetts Institute of Technology (MIT), the University of Michigan, the DOE Argonne National Laboratory (ANL), the DOE Los Alamos National Laboratory (LANL), the National Oceanic and Atmospheric Administration (NOAA) Geophysical Fluid Dynamics Laboratory (GFDL), and the NOAA National Centers for Environmental Prediction (NCEP). Recently the U.S. Navy, U.S. Air Force, and U.S. Army joined the ESMF collaboration through the creation of the Battlespace Environments Institute, a virtual center that will couple codes from multiple services to create integrated forecasts.

The ESMF software consists of a superstructure for coupling and exchanging data between component models (e.g. atmosphere, ocean) and model subcomponents (e.g. physics, dynamics), and an infrastructure consisting of (1) data structures for representing grids and fields and (2) an optimized, portable set of low-level utilities. The data constructs and low-level utilities are used by the coupling superstructure and may also be used separately to compose scientific applications. Conceptually, an application running under ESMF may be thought of as a sandwich, with the upper coupling layer and lower utility layer provided by ESMF and the middle layer provided by the application developer. The ESMF superstructure sits above the components of an application, controlling inter-component data transfer and sequencing. The ESMF infrastructure lies below the components, offering integrated tools for intra-component communication, error handling, time management, profiling, and other standard modeling functions. More information on ESMF can be found at [4,5].

To examine and test ESMF capabilities, we have specifically constructed several experiments involving model components from different organizations. In particular, we have coupled the NASA–NCAR finite-volume Community Atmosphere Model (fvCAM) [6] with the NCEP Spectral Statistical Interpolation (SSI) analysis [7], NASA ARIES atmospheric model [8] with NCEP SSI, and the GFDL Flexible Modeling System (FMS) B-grid atmosphere model [9] with the MITgcm ocean model [10]. These experiments illustrate the ESMF component strategy and coupling services.

ESMF COMPONENT MODEL AND COUPLING UTILITY

Weather and climate models are natural applications for component technology. In general, a climate model consists of four primary geophysical components: atmosphere, ocean, land surface, and

DOI: 10.1002/cpe

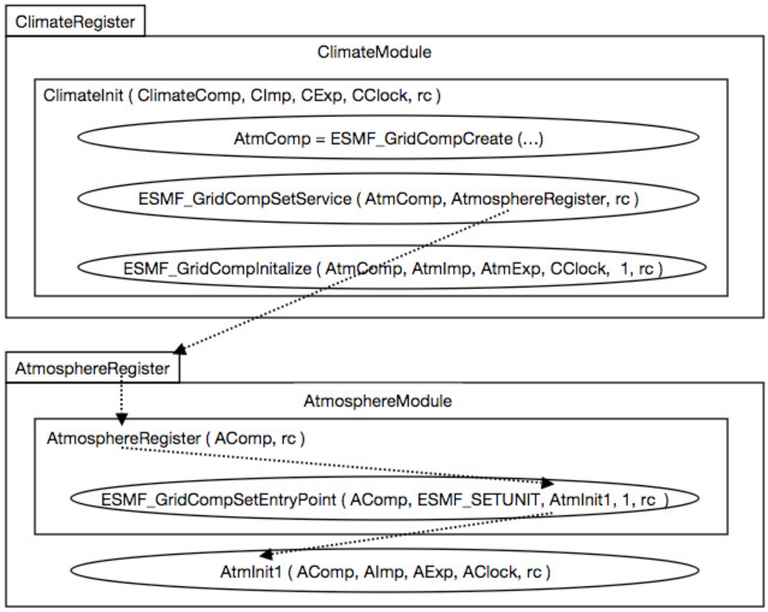


Figure 1. Schematic of the ESMF registration process.

sea ice. An atmosphere component is often further decomposed into separate dynamics and physics components. Within the physics parameterization component, each subgridscale process, such as radiative transfer and cloud physics, can also be encapsulated into a component.

ESMF recognizes the nature of weather and climate models and deploys component technology. In the following, we briefly describe the ESMF component and coupling utility from the user's perspective. More detailed information on design and implementation can be found in [4,11].

(1) Each ESMF component, a Fortran 90/95 module, has to implement three standard function interfaces: Initialize(), Run(), and Finalize(). ESMF provides register utilities (SetServices() and SetEntryPoint()) to enable a user to map the corresponding routines of their component to these three functions (i.e. change routine names), as shown in Figure 1. The SetServices terminology was taken from the CCA project. In CCA, setService is used for a component to register its ports to the CCA framework, which is different from ESMF's [2,12].
The rationale behind ESMF registration utilities is that the models that ESMF intends to support in the first stage are production-quality codes. Those models have been developed by many researchers over a decade or so, designed in a functional rather than object-oriented style, and coded mostly in Fortran 77 with some in Fortran 90/95. So, the first priority is to find a way of uniting various function interfaces so that one model can use the functions of another model in a general way. By defining three standard function interfaces, Initialize(), Run(), and Finalize(), and deploying a C functional pointer table [4,11], ESMF realizes the CCA's setService in a

DOI: 10.1002/cpe

style of static rather than dynamic binding for Fortran subroutines. In that sense, ESMF uses convection rather than programming language features to achieve the limited (manual and static) generic features.

Figure 1 illustrates how the ESMF registration is used by an ESMF component to make its functions available for another component to use. In a model such as an atmosphere model, the routines can be sorted into three kinds of subroutines: initialization, run, and finalization. A user first creates a Fortran module (e.g. AtmosphereModule) and makes the register subroutine name (e.g. AtmosphereRegister) public so that other Fortran modules such as ClimateModule can access it. (The register subroutine is similar to CCA's providePort. Three ESMF standard function interfaces act like CCA's usePort.) Then inside that registration subroutine (e.g. AtmosphereRegister), the user needs to add the name of the subroutine (e.g. AtmInit1) to the argument list of ESMF_GridCompSetEntryPoint(). Typically, there are also several initialization subroutines for initial conditions, boundaries, etc. In that case, ESMF provides the feature of 'phase' to allow each initialization subroutine to add its subroutine name to the argument of ESMF_GridCompSetEntryPoint() with the corresponding phase. For an atmosphere component with two phase initializations, and a single phase run and finalize, the relevant code is as follows:

```
Public AtmosphereRegister

Subroutine atmosphereRegister(AComp, rc)
```

where AComp is an object of ESMF GridComp derived type. This object carries all of the data needed for this component except import state (e.g. AImp), export state (e.g. AExp), and clock (e.g. AClock). 'rc' is an integer indicting whether this subroutine is called successfully or not.

```
call ESMF_GridCompSetEntryPoint(AComp, ESMF_SETINIT,
AtmInit1, 1, rc)

call ESMF_GridCompSetEntryPoint(AComp, ESMF_SETINIT,
AtmInit2, 2, rc)

call ESMF_GridCompSetEntryPoint(AComp, ESMF_SETRUN, AtmRun,
ESMF_SINGLEPHASE, rc)

call ESMF_GridCompSetEntryPoint(AComp, ESMF_SETFINAL,
AtmFinal, ESMF_SINGLEPHASE, rc)
```

When an ESMF component (e.g. ClimateModule) or driver uses the subroutines of initialization, run, and finalization of this atmosphere model, it will deploy the Fortran 90/95 'Use' feature to link the register subroutine (e.g. AtmosphereRegister) through its module (AtmosphereModule). Then the register subroutine name (e.g. AtmosphereRegister) is added into the component object representing the atmosphere model (e.g. AtmComp) by calling ESMF_GridCompSetService(). After that, those subroutines can be used through this component object (e.g. AtmComp). The relevant code is as follows:

```
Use AtmosphereModule, only: AtmosphereRegister

Type (ESMF_GridComp), save::AtmComp

call ESMF_GridCompSetServices(AtmComp, AtmosphereRegister,
rc)

call ESMF_GridCompInitialize(AtmComp, AtmImp, AtmExp,
CClock, 1, rc=rc)

call ESMF_GridCompInitialize(AtmComp, AtmImp, AtmExp,
CClock, 2, rc=rc)

call ESMF_GridCompRun(AtmComp, AtmImp, AtmExp, CClock,
rc=rc)

call ESMF_GridCompFinalize(AtmComp, AtmImp, AtmExp,
CClock, rc=rc)
```

(2) There are four arguments in each standard function interface: component object, import state, export state, and clock. The component object contains adequate information about the component, such as grid type, processor layout, and functional pointer table, supporting the register utility. Import and export states contain fields to be transferred between components. The clock is used for advancement and synchronization of components. In addition, an internal state is provided to pass the internal supporting data to Initialize(), Run(), and Finalize().
(3) Components interact only through the import and export states. The import and export states are of ESMF_State type. ESMF_State is a Fortran 90/95-derived type and is the only way for components to exchange data. This Fortran-derived type can contain other ESMF data types: ESMF_Array, ESMF_Field, and ESMF_Bundle. An ESMF_Array contains a data pointer, and information about its association data type, precision, and dimension. An ESMF_Field consists of ESMF_Array and its underlying grid, ESMF_Grid. An ESMF_Bundle is a group of ESMF_Fields on the same ESMF_Grid. To enable an ESMF coupler to support multiple components, an ESMF_State can also contain the data of other ESMF_States. For example, an export state can contain several sub export states. Each sub export contains the output data for its destined components. ESMF_State is implemented as a container allowing a user to link their native Fortran 90/95 array to an export state and extract arrays and grid information from an import state.
(4) ESMF component does not support inheritance. However, an ESMF component can be written into a nested component (e.g. ClimateModule) where ESMF_GridCompSetEntryPoint() is deployed to provide its functions (e.g. ClimateInit) to the upper-level component and ESMF-GridCompSetService() to use the functions (e.g. AtmInit1) from the lower-level component (AtmosphereModule).

Although the level of abstraction in this component definition is not as high as that in CCA, it is easy for a production-quality code written in Fortran to interact with other codes written in Fortran

DOI: 10.1002/cpe

in a standard manner. For example, NCEP's SSI analysis system has been developed and modified by many developers over more than 10 years. One difficulty for coupling is that the code structure is not well organized in the areas of import/export data and initialization routines. These are three of the most time-consuming tasks encountered in making the SSI code ESMF-compliant. When the SSI code becomes ESMF-compliant, coupling with the atmospheric models, such as NASA–NCAR fvCAM or NASA ARIES, is significantly simplified.

Climate model components often use different numerical grids and resolutions. Therefore, regridding services are frequently needed in model coupling. Since operational climate models are implemented in parallel, a parallel regridding utility is highly desirable. The ESMF team took a popular serial regridding package, the Spherical Coordinate Remapping and Interpolation Package (SCRIP) [13], made its algorithm parallel, and implemented it with the ESMF parallel communication library. This approach gave ESMF the capabilities of computing the regridding weights and determining the communication patterns needed at execution time. The reason for paralleling SCRIP with ESMF parallel communication library rather than MPI, PVM, and OpenMP is that a user can perform regridding operations with ESMF_Field, which is more friendly to an ESMF user.

In the following we give brief descriptions of one ESMF-compliant component and the three ESMF interoperability experiments.

NCEP SSI ANALYSIS

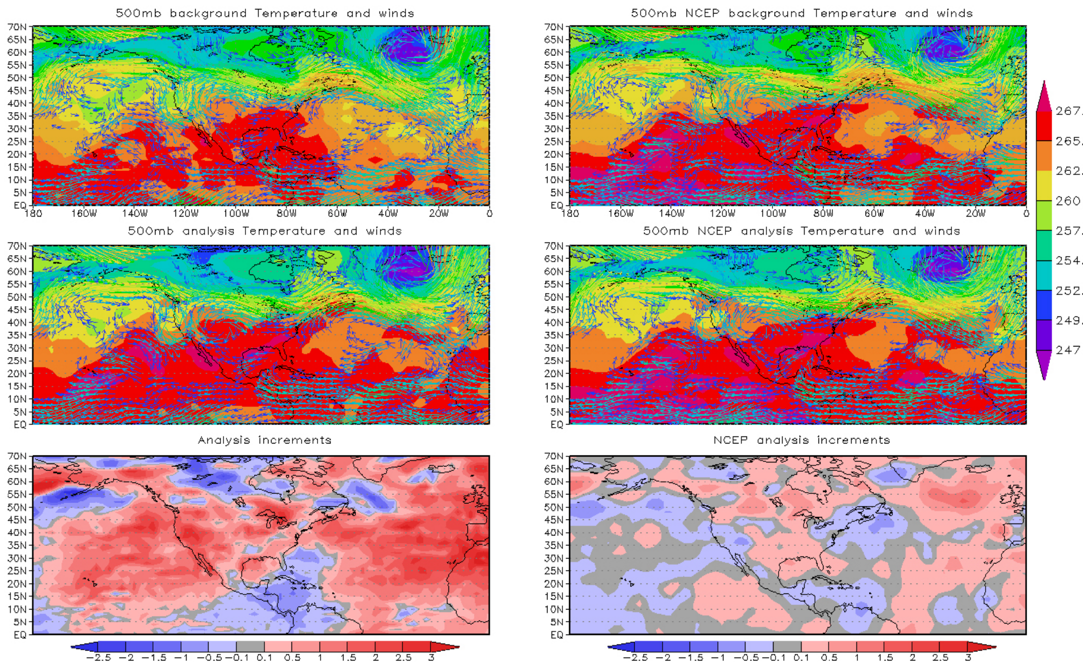
To create the ESMF-compliant version of an existing model or data assimilation system, we typically need to restructure the code to resemble an ESMF-compliant grid component. That implies creating Initialize(), Run(), and Finalize() routines and specifying import and export states and the clock. After that, we wrap the code with ESMF routines to create an ESMF-compliant version. Finally, we add the desired ESMF utilities such as Configure and Log Error to further improve its standardization and capabilities such as debug.

In restructuring the code of NCEP SSI analysis system, we have specifically carried out the following steps.

(1) Extract all of the data to be exchanged with other ESMF components as well as out of the original I/O system and create the import/export states.
(2) Create an internal state that contains all of the important parameters, indexes, and other data arrays to support Initialize(), Run(), and Finalize().
(3) Change all of the original control parameter input or read methods, such as reading a namelist or the header of a huge input data disk file, to ESMF Configure input file format.
(4) Reorganize the original code into the three routines Initialize(), Run(), and Finalize(). Move non-repeated computations of the original code into the Initialize() routine to improve the performance.

These steps are very time-consuming and require considerable domain-specific expertise. However, creating an ESMF-compliant component greatly increases the interoperability of a production-quality legacy code, such as SSI, with the Earth system models of other organizations. That advantage is illustrated in two of the three cross-organization interoperability experiments described below.

DOI: 10.1002/cpe



(a) (b)

Figure 2. The NASA–NCAR atmosphere model (fvCAM) coupling with the NCEP data analysis system (SSI). (a) Temperature in absolute temperature (K) and wind output (m s^{-1}) from the fvCAM first guess (top), the fvCAM–SSI coupled system (middle), and then the difference (bottom). (b) The same quantities for the operational NCEP Global Atmospheric Model–SSI data assimilation system. Note that both systems used the same initial conditions.

NASA–NCAR fvCAM WITH NCEP SSI ANALYSIS

In this interoperability experiment, the NASA–NCAR fvCAM is coupled to the NCEP SSI analysis system for the first time. The fvCAM configuration uses a $144 \times 91 \times 26$ computational grid in physical space, while SSI uses a $192 \times 96 \times 28$ grid in spectral space, so-called T62. To couple with fvCAM in physical space, SSI was wrapped with a Gaussian grid for importing and exporting data.

In addition to making the fvCAM and SSI ESMF components, we also needed to construct two ESMF coupler components. These two couplers perform regridding in physical space as well as physical unit conversion. In a production-quality model, it is not unusual to find that the operations of importing/exporting data, initialization, and run are not clearly organized. The ESMF component model (three standard function interfaces: Initialize(), Run(), and Finalize() with the standard data exchange format, ESMF_State) promotes standardization and greatly facilitates cross-organization coupling.

As shown in Figure 2, the newly constructed data assimilation system based on the coupled fvCAM–SSI delivers data assimilation results that are comparable to the operational NCEP data assimilation system.

DOI: 10.1002/cpe

This interoperability experiment suggests that ESMF can facilitate comparing and using the models from different organizations for research and operational goals.

NASA ARIES GCM WITH NCEP SSI ANALYSIS

This interoperability experiment consisted of coupling the NASA ARIES [5] with NCEP's operational analysis system (SSI). The model variables are defined in physical space on a 144×90 horizontal grid resolution corresponding to a $2.5° \times 2°$ grid. ARIES uses a Lorenz grid in the vertical with 34 sigma levels extending from the surface and extrapolated to 0 hPa. The coupling between ARIES and SSI was identical to the coupling between fvCAM and SSI except for the different geophysical grids and contents of import/export states for the global climate models. The couplers between two components perform the conversion or transformation of meteorological fields. The ARIES–SSI system is a simplified weather forecast system whose main objective is to make use of ESMF. The analysis of performance overhead due to the ESMF component layer on SSI is insignificant as reported in [11]. The data volume for this application uses 301 400 observations per synoptic cycle.

Figure 3 shows that the newly constructed data assimilation system based on the coupled ARIES–SSI also delivers comparable data assimilation results that are comparable to the operational NCEP data assimilation system. It is expected that the difference in the results can be reduced after tuning the ARIES model.

The coupler between ARIES and SSI was developed based on that between fvCAM and SSI. It took considerably less time compared to the fvCAM–SSI coupler, which shows the generic and reusable features of the ESMF coupling utility.

GFDL FMS B-GRID ATMOSPHERE WITH MITgcm OCEAN

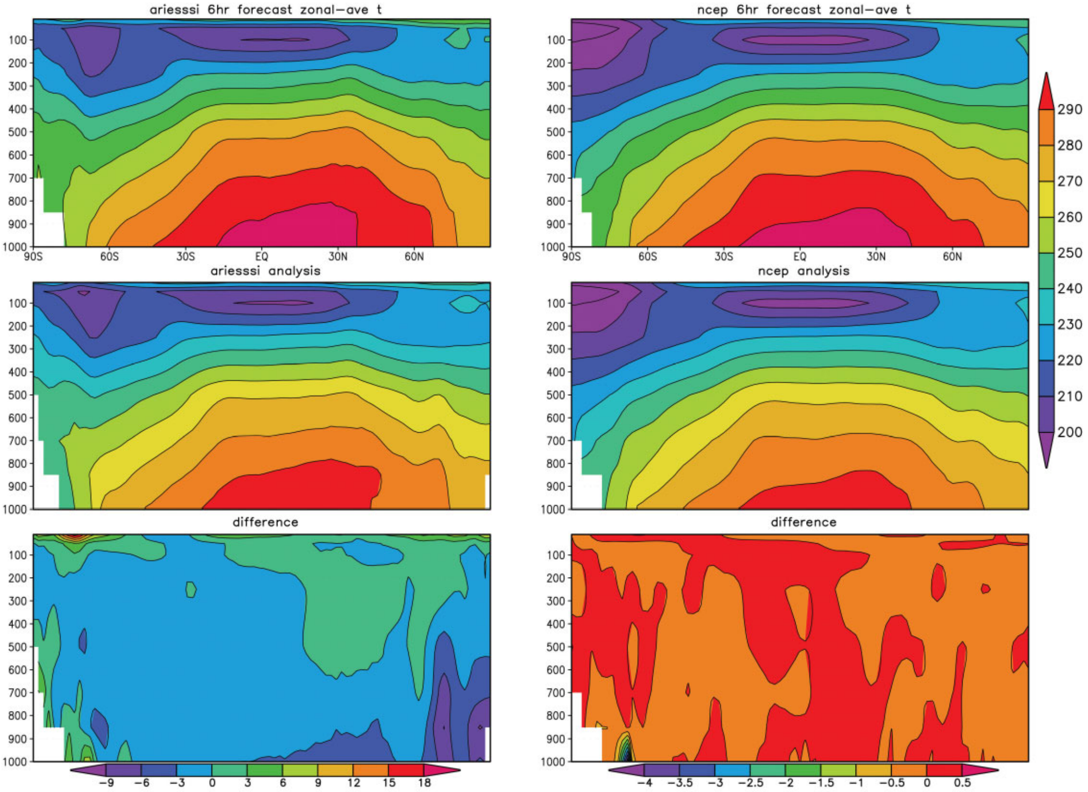
In this experiment, the GFDL FMS B-grid atmosphere (plus land and ice) component is configured with a 2° latitudinal resolution and 2.5° longitudinal resolution, resulting in a computational grid that is 144×90 grid points in the horizontal. It is coupled to a MITgcm ocean component that uses 2.8125° resolution both zonally and meridionally. The atmosphere component is decomposed into 30 longitude bands. The MITgcm component is decomposed into 30 latitude circles.

As in the fvCAM–SSI interoperability experiment, the atmosphere and ocean models were first converted into ESMF components. Then, two ESMF coupler components were created. These components transform physical units and regrid physical grid variables. In the couplers, the ESMF parallel regridding utility was employed. It is interesting to note that the coding style of the coupler is very similar to that of the fvCAM–SSI coupler, although the fvCAM–SSI and the GFDL B-grid atmosphere-MITgcm ocean coupled systems are quite different from a scientific viewpoint. In that sense, the experiments show the potential for ESMF to achieve the goal of standardizing model coupling procedures and facilitating coupling.

The coupled system was run for five coupling cycles. The smooth evolution of sea surface temperature as a function of time shows that the coupled system constructed with ESMF was stable during the brief simulation period (see Figures 4(a) and (b)). The longer running time was not performed since a stable coupled system also depends on the scientific coupling mechanism, which is beyond the scope of this cross-organization interoperability experiment aimed for software capability.

However, the ESMF coupling mechanism has been demonstrated to be stable over a few hundred time steps in the cross-organization model couplings between UCLA atmosphere and MITgcm ocean

DOI: 10.1002/cpe



(a) (b)

Figure 3. The NASA ARIES atmospheric model coupled with NCEP's analysis system (SSI). (a) The zonally averaged temperature six-hour forecast from ARIES (top), the subsequent analysis (middle), and then the difference (bottom). (b) The same quantities for the operational NCEP Global Atmospheric Model–SSI data assimilation system. Note that both systems used the same initial conditions.

models, as well as the UCLA atmosphere and LANL POP ocean models [14]. These two coupling experiments have shown that ESMF provides a software infrastructure to compare the ENSO prediction skills of models. In fact, the same ESMF-compliant MITgcm component was used in these two interoperability experiments.

CONCLUSION

Through three interoperability experiments consisting of never-before coupled models, assembled with production-quality weather and climate models from different groups, we show that the ESMF component model and coupling services can facilitate model coupling across organizations. The lesson learned is that data types, function interfaces, and coupling mechanisms derived from existing software

DOI: 10.1002/cpe

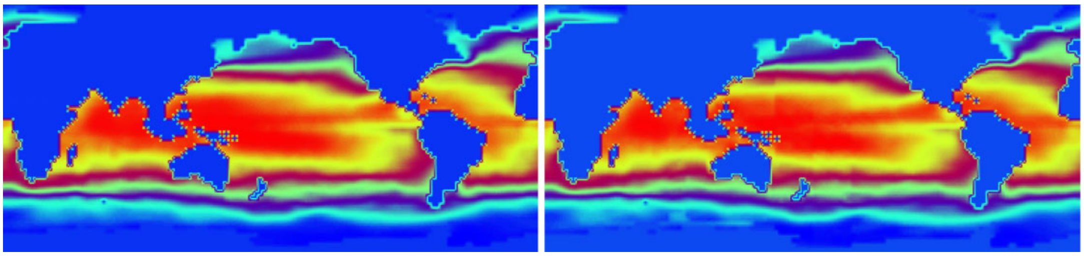

(a) (b)

Figure 4. The GFDL FMS B-grid atmosphere model coupled with the MITgem ocean model. (a) Initial global sea surface temperatures in degree Celsius (°C). (b) Global sea surface temperatures after five iterations.

frameworks and packages out of the user community with adequate user inputs starting in the early stage soften the learning curve and keep the performance degradation tolerable.

ACKNOWLEDGEMENTS

We thank Cecelia DeLuca, Jon Wolfe, Tom Clune, and Mike Seablom for reading the manuscript and offering helpful suggestions, and Jarrett Cohen for editing the manuscript and figures.

REFERENCES

1. Talbot B, Zhou SJ, Higgins G. Review of current frameworks. *Technical Memorandum NASA/TM-2001-20992*, NASA, 2001.
2. Zhou S. Coupling climate models with Earth System Modeling Framework and Common Component Architecture. *Concurrency Computation: Practice and Experience* 2006; **18**(2):203–213. DOI: 10.1002/cpe.912.
3. McInnes LC *et al*. Parallel PDE-based simulations using the Common Component Architecture. *Numerical Solution of Partial Differential Equations on Parallel Computers*, Bruaset AM, Bjorstad P, Tveito A (eds.). Springer: Berlin, 2006.
4. ESMF. http://www.esmf.ucar.edu [17 August 2006].
5. Hill C, DeLuca C, Balaji V, Suarez M, da Silva A, the ESMF Joint Specification Team. The architecture of the Earth System Modeling Framework. *Computing in Science and Engineering* 2004; **6**(1):18–28.
6. fvCAM. http://www.ccsm.ucar.edu/models/atm-cam/ [17 August 2006].
7. Kanamitsu M. Description of the NMC Global Data Assimilation and Forecast System. *Weather and Forecasting* 1989; **4**:335–342.
8. Suarez MJ, Takacs LL. Documentation of the ARIES/GEOS Dynamical Core, Version 2. *Technical Memorandum 104606*, vol. 5, NASA, 1995.
9. GFDL Flexible Modeling System. http://www.gfdl.noaa.gov/∼fms [17 August 2006].
10. MITgcm. http://mitgcm.org [17 August 2006].
11. Collins N, Theurich G, DeLuca C, Suarez M, Trayanov A, Balaji V, Li P, Yang W, Hill C, da Silva A. Design and implementation of components in the Earth System Modeling Framework. *International Journal of High Performance Computing Applications* 2005; **19**(3):341–350.
12. Common Component Architecture. http://www.cca-forum.org/ [17 August 2006].
13. Jones PW. First and second order conservative remapping scheme for grids in spherical coordinates. *Monthly Weather Review* 1999; **127**:2204–2210.
14. Mechoso CR, Boezio GC, Menemenlis D. Experimental ENSO predictions by the UCLA atmospheric GCM coupled to the MIT and POP Oceanic GCMs using the Earth System Modeling Framework (ESMF). *Proceedings of the NASA Earth Science Technology Conference 2005*, College Park, MD, 28–30 June 2005. Available at: http://www.esto.nasa.gov/conferences/estc2005/author.html [17 August 2006].